# Automated Hex Meshing For Turbomachinery Secondary Air System

Feng Wang[1] and Luca di Mare[2]

[1] Department of Mechanical Engineering, Imperial College London, UK
feng.wang207@ic.ac.uk
[2] Department of Mechanical Engineering, Imperial College London, UK
l.di.mare@ic.ac.uk

**Abstract:** In this paper, we present a novel process of creating hexahedral meshes for the turbomachinery secondary air system. The meshing process is automated from the geometry import to the mesh setup and requires minimum human interventions. The core of the process is a hexahedral meshing algorithm with a boundary layer mesh automatically created. The hex meshing algorithm combines the pave-sweep, general sheet-insertion and a novel technique which creates the boundary layer mesh by carefully placing, maintaining and dicing a buffer layer around a geometry. After the mesh is created, relevant boundary conditions for the mesh are also assigned automatically. The whole meshing process is systematically automated and has the potential to considerably reduce the time cost in meshing the turbomachinery secondary air system.

## 1 Introduction

Computational Fluid Dynamics (CFD) has found wide applications in the turbomachinery industry. Over the years, turbomachinery CFD has reached the stage of embracing simulations of real 3D geometries for design purposes [1, 2]. This has posed great challenges to geometry modeling, mesh generation and flow solvers. Within the design environment numerous design variations need to be evaluated in a timely manner, this requires that geometries are convenient to edit and mesh generators are able to respond to the changes in geometries quickly, create the mesh rapidly and set up the mesh as automatically as possible. The purpose of this paper is to present our effort of addressing these needs.

The meshing process can be divided into three sub-processes:

- Geometry extraction and cleanup
- Mesh generation

- Mesh setup

Geometry models are normally represented explicitly by boundary representation(BREP), such as NURBS patches, edges, etc. BREP is widely used in CAD systems and is a standard input to numerous mesh generators for CFD, such as ANSYS ICEM CFD, GAMBIT, etc. However, typical BREP geometries are frequently "dirty" due to numerical problems, imprecise design or data exchange issues, which requires a considerable amount of effort to clean up so that mesh generators can perform satisfactorily. Furthermore, the geometries might have heterogeneous sources, and this could lead to difficulties in modifying the geometries and possibly introduce inconsistencies. Last but not the least, BREP is simply a collection of geometrical entities and unable to automate the mesh setup process, e.g. by aiding the automated assignment of boundary related conditions. Therefore, a favorable geometry should be *inherently clean, easy to import and edit, and facilitate the mesh setup*. For this purpose, we have recently developed a system for representing the geometries of complex assemblies such as turbomachinery in a way which facilitates the construction of new and possibly unorthodox configurations from scratch whilst facilitating the automation of analysis procedures [1].

The most common types of elements created by 3D automatic meshing algorithms are tetrahedron [3, 4, 5] and hexahedron. Hexahedral elements are generally preferred due to their efficiency and accuracy in the computational process [6]. Currently there are several popular hexahedral meshing algorithms, such as the multi-block structured method [7], the octree-based method [8, 9] and the pave-sweep method [10]. Among them, the pave-sweep method can produce high quality, body-fitted, topologically conformal hexahedral meshes and close to the boundaries the mesh contours can be aligned with the boundary contours satisfactorily. Augmented with general hexahedral sheet (hex sheet) insertion methods [11], the pave-sweep method is capable of producing high quality hex meshes for complicated 3D geometries. For CFD the generation of a boundary layer mesh close to viscous walls is necessary in order to capture the high gradient close to the walls. For the multi-block structured method, mesh lines are clustered to viscous walls by solving partial differential equations [12] but this is only limited to relatively simple geometries. Even though the octree-based method is able to create the boundary layer mesh for a wide class of geometrises by the level-set method [9], the mesh contour outside of the boundary layer mesh normally does not follow the boundary contours. For the pave-sweep method, the sheet insertion method can be used to create a boundary layer mesh but there are few papers presenting a detailed approach of automatically creating a boundary layer mesh for a 3D geometry.

Mesh setup is normally not considered as part of the meshing process. However, if considering the complex turbomachinery secondary air system(SAS), one could spend a large fraction of time to set up the mesh. One needs to identify which part of the geometry is a free-stream boundary, which part is

rotating at what speed, which part is stationary, etc. This process is tedious, time-consuming and error-prone. Furthermore, within a design environment, once the geometry is modified and meshes need to be rebuilt, all the procedures need to be repeated. Therefore, mesh setup should also be automated and considered as part of our meshing process.

Despite the advances in hex meshing algorithms, there is a lack of a systematic and automated meshing process which creates a quality hexahedral mesh with a boundary layer mesh and facilitates the SAS design. The research described herein is to meet this need. In this paper, we present a hex meshing process which is automated from geometry import to mesh setup. The core of the process is a hex meshing algorithm which is based on the pave-sweep and general hex sheet insertion methods. A boundary layer mesh is also automatically created by dicing a 3D buffer layer.

The rest of the paper is organized in the following manner: section 2 briefly explains the geometry database we have developed recently, section 3 gives the outline of the meshing process, section 4 demonstrates the method to extract a cavity from our geometry database, section 5, 6, 7 and 8 illustrate the hex meshing algorithm to create a quality hexahedral mesh with a boundary layer mesh, section 9 shows an example mesh generated by the meshing process and conclusion and future work follow in section 10.

## 2 The General Assembly

The current work is conducted within our Virtual Engine(VE) environment [1], which is an objected-oriented design and analysis systems for gas turbines and can conduct both low-fidelity and large-scale high-fidelity simulations. The design of VE is based on abstraction and it adopts tree-like geometrical features and a tree of functional models. For the geometrical features, which is termed the General Assembly(GA), its purpose is to develop a system for representing the geometries of complex assemblies in a way which facilities the construction of new and possibly unorthodox configurations from scratch and, at the same time, facilitates the automation of analysis procedures. The geometry is represented in a parametric way, so as to allow changes in one component to be propagated through the assembly whilst maintaining mechanical and kinematic consistency. In support of high-fidelity analysis, the system can turn the parametric description of an assembly into a set of geometric primitives in 2D and 3D by Boolean operation, such as polygons and polyhedra respectively.

We describe the whole engine as a generic feature and consider its main components, the spools and dressing as its child features; each blade is treated as a child feature of the spools; all the mechanical parts that are associated with the blades are considered as child features of the blades, and so on. Therefore, starting at the root, namely the engine, we populate the feature tree all the way down to every detail of an engine, this is illustrated in Figure 1.

In order to represent the SAS, special types of features are made to represent orifices, holes,slots etc. These features modify the basic geometry of the engine components by removing material, hence creating fluid connection between different parts of the assembly. All the geometric primitives describing the geometry of an engine are tagged with unique identifiers, this allows every line or surface to be associated with a feature and facilitate the automation of the meshing process as well as automatic attribution of boundary conditions and boundary treatments.

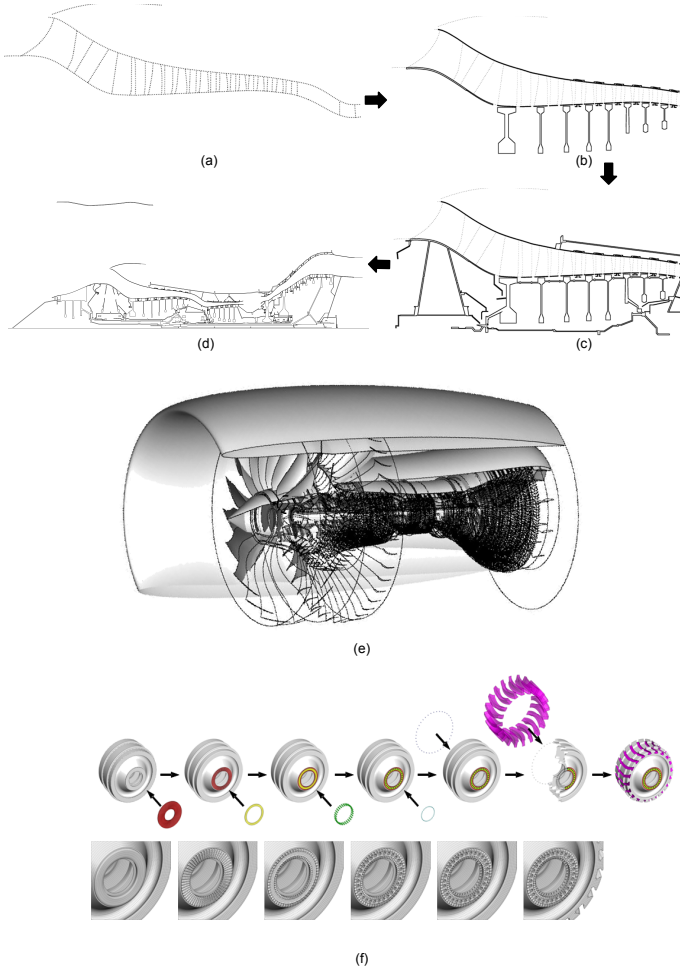


**Fig. 1.** Assembling process for 2D and 3D GA. (a)-(d) illustrate the formation of a 2D GA. (e) shows the 3D GA and (f) illustrates the formation of a 3D fan disc.

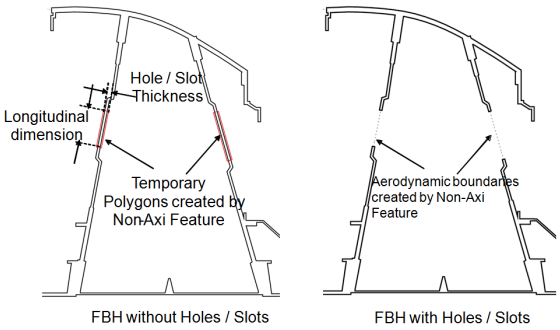


**Fig. 2.** Illustration of creating a SAS cavity.

Even though the physical shapes of components might change with different operating conditions, their nature stay the same(For example, a blade is always a blade), therefore, it is natural to associate boundary conditions to geometric features and to recognize that each geometric feature will hold a certain set of data items but with possible different values at different operating conditions.

Based on what we have described above, we can see that the GA can be used with advantages within the meshing process by guaranteing geometric consistencies, thus removing the need for a geometry cleanup, or by allowing automatic association of boundary conditions. Therefore, the GA is a well-qualified candidate for our requirement of being inherently clean, easy to import and edit, and facilitate the mesh setup.

## 3 Outline of The Meshing Process

We now illustrate the meshing process by meshing a cavity with connecting orifices, which represents the most common and general case in the SAS. The whole process is made up of the following steps:

1. Extract a cavity from the GA.
2. Pave the 2D buffer layer and create an offset boundary for the cavity.
3. Create a quadrilateral mesh in the region enclosed by the offset boundary of the cavity. Combine the quad mesh with the buffer layer and create a hexahedral mesh with a 3D buffer layer by sweeping the quad mesh.
4. Graft non-axi-symmetric orifices.
5. Dice the 3D buffer layer to create the boundary layer mesh.

## 4 Geometry Extraction

As is described in Section 2, the GA can turn itself into a set of 2D primitives, i.e. polygon. We can then interpret the GA as a mere set of non-self-intersecting polygons and apply a slitting procedure along a set of selected lines. The result is a set of polygons which have:

- negative areas: these polygons represent air system cavities
- positive areas: these polygons always have sides on the walls wet by the main gas path, the bearing cavities or the engine centerline. These polygons always correspond to what the air-system sees as boundary conditions.

The basic ingredients for extracting a cavity from the GA are a set of aerodynamic boundaries, available as a set of lines intersecting the GA polygon and by selecting different sets of aerodynamic boundaries one can extract a specific part of the SAS. Figure 3 shows all the cavities extracted by the slitting procedure and Figure 4 (left) shows the extracted cavity by four aerodynamic boundaries: 1) FHOLE, 2) FCLR, 3) DHOLE, 4) SHOLE. After the cavity is extracted, the non-axi-symmetric holes, slots and orifices that are present in the cavity are temporarily removed. Figure 4 shows the processed cavity and also shows the physical boundary information associated with the cavity.

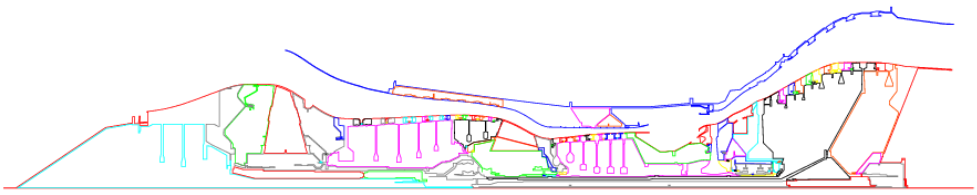

**Fig. 3.** Extracted cavities for a three-shaft engine by the slitting procedure

## 5 Paving The Buffer Layer

The purpose of the buffer layer is to form several sheets of quadrilaterals (quad sheets) to capture the boundary and provide the offset boundary. The method to create the buffer layer is similar to the popular paving algorithm to create unstructured quadrilateral elements [13] and hence the thickness of the resulting buffer layer depends adaptively on the local grid spacing on the boundary. The main difference is that row adjustment, such as wedge insertion and tuck formation, is suppressed, because we want to avoid creating unnecessary quad sheets on the boundary so as to enhance the regularity of the buffer layer and the quality of its resulting boundary layer mesh.

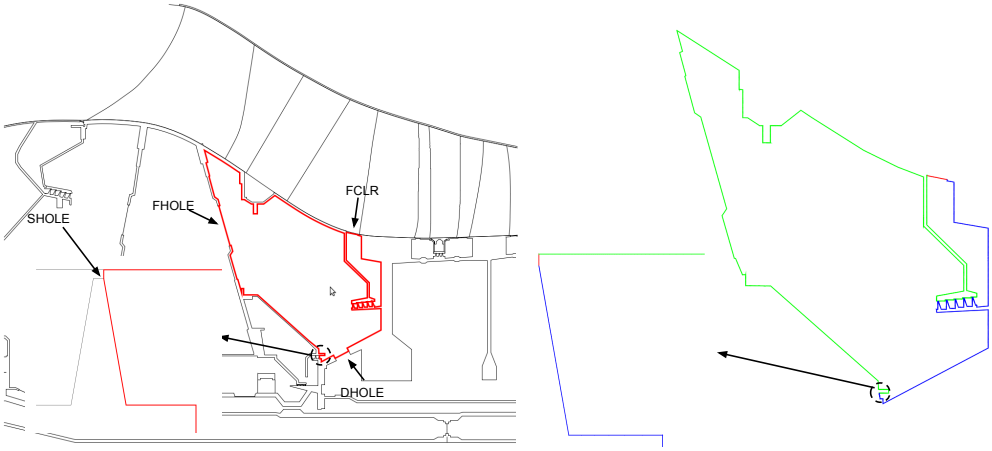


**Fig. 4.** Left: Extraction of a cavity. Right: Cavity with removed non-axi-symmetric holes and boundary information provided by the GA. The Green represents the metal that is stationary, the blue represents the metal that is rotating with the intermediate pressure(IP) spool and the red shows axi-symmetric slots.

Similar to the paving method, based on the interior angle(see Figure 5) on the boundary, each node on the boundary is classified as row end node, row side node, row corner node and row reversal node, as is shown in Table 1. Tucks and wedges are inserted only based on the type of a boundary node, row collision is detected and seaming is also performed to avoid overlapped quadrilaterals for geometries with small input angles. Figure 5 shows the schematic of the paved buffer layer and offset boundary based on the boundary node types.

**Table 1.** Classification of boundary nodes and corresponding interior angle ranges.

| Node type | Angle range |
|---|---|
| Row end node | $[0^o, 135^o)$ |
| Row side node | $[135^o, 225^o)$ |
| Row corner node | $[225^o, 315^o)$ |
| Row reversal node | $[315^o, 360^o)$ |

## 6 Creating Quadrilateral and Hexahedra Mesh

After the buffer layer and the offset of the boundary are created, the rest of the region is filled with quadrilateral elements. We have chosen to implement

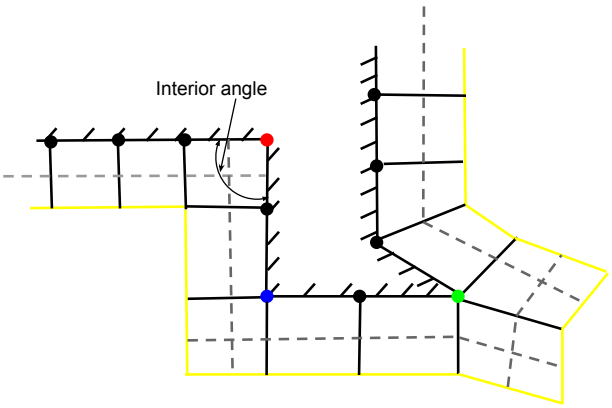


**Fig. 5.** Boundary node classification: row side node(Black), row end node(Red), row corner node(Blue), row reversal node(Green), and their respective templates of constructing a buffer layer. The grey dash lines represent the quad sheets and the yellow line represents the offset boundary.

the Q-morphing [14] algorithm, as mesh contours of the resulting quadrilateral mesh (quad mesh) exhibit a good boundary alignment close to the boundary. The source triangulation is created by Delaunay refinement with an off-center point-insertion scheme [15], because it tends to produce fewer elements than a standard Delaunay refinement or an advancing front method, and the resulting quad mesh size is correspondingly smaller. The quality of the quadrilateral mesh is improved by local [16] and non-local [17] topological cleanup followed by several iterations of angle-based smoothing [18]. A complete quad mesh for the cavity is then formed by combining the buffer layer and the quad mesh created by Q-morphing. The resulting mesh is treated as representing the average projection of the cavity on the meridional plane($X - Z$ plane) and a first hex mesh with a 3D buffer layer is created by sweeping the quad mesh around the $X$ axis, as is illustrated in Figure 6.

## 7 Grafting Non-Axi-Symmetric Features

In the hex mesh created by sweeping, non-axi-symmetric features are missing and they are added by grafting. For consistency, we adopt the grafting terminology used in [19], as is summarized in Table 2. Grafting contains three steps: 1) creating conformal graft surfaces, 2) improving the mesh quality inside and outside graft loops, 3) extruding branches. The methods to make a conformal graft surface and extrude branches are similar to [19], here we only describe our method of improving the mesh quality inside and outside graft loops and

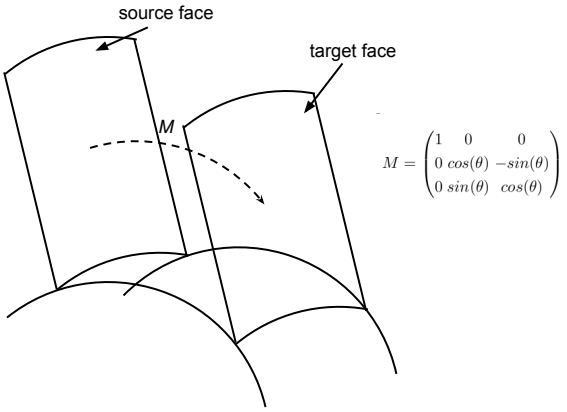


**Fig. 6.** Schematic of creating a hex mesh by spinning. $M$ is the rotation matrix.

without losing generality, it can be demonstrated by a simple example. as is shown in Figure 7 (left).

**Table 2.** Terminology in grafting

| Term | Description |
|---|---|
| Trunk | A hex mesh where branches are grafted on. |
| Branch | A missing non-axi-symmetric feature. |
| Base surface | A surface mesh on which a branch is grafted on. |
| Graft surface | The intersection of a base surface and a branch. |
| Graft loop | Boundary edges of a conformal graft surface. |

We first define the smallest convex region, $\Omega$, on the base surface that is intersected by a branch. If the trunk is meshed by sweeping, then $\Omega$ is represented by a subset of the structured grid on the base surface, this is illustrated in Figure 7(right, highlighted by dark red). When the graft surface is made to be conformal to the branch, $\Omega$ is separated into two sub-regions, $\Omega_0$ and $\Omega_1$, one is inside the graft loop and one outside the graft loop respectively. These two sub-regions are treated as two separate shrink sets and two corresponding hex sheets are then inserted to improve the mesh quality inside and outside the graft loop. Figure 8(a) shows $\Omega_0$ and the inserted hex sheet to improve the mesh quality inside the graft loop. Figure 8(b) shows $\Omega_1$ and the inserted hex sheet to improve the mesh quality outside the graft loop. Figure 8(c) shows the two inserted hex sheets are inflated after local surface and volume smoothing and Figure 8(d) shows the final hex mesh for the example geometry.

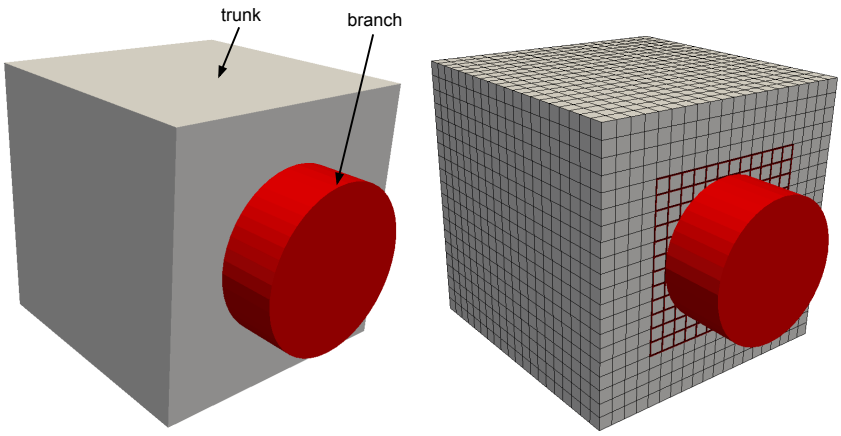


**Fig. 7.** An example geometry for grafting.

As we can see, similar to the typical procedure in grafting, we place a hex sheet to improve the mesh quality inside the graft loop. However, compared with typical ways of improving the mesh quality outside the graft loop, which pillows low quality elements [20] immediately outside the graft loop or insert a global hex-sheet [11], our approach has several advantages. The 3D buffer layer, which we have carefully placed, is only subject to local modifications. As is illustrated in Figure 9(a), the grey dash grey line represents a hex sheet of the 3D buffer layer and after we insert the two hex sheets, it is not corrupted and only its curvature is influenced locally close to the graft loop, therefore, the 3D buffer layer is well preserved. This is a crucial property since the 3D buffer layer will be used to create the 3D boundary layer mesh. Based on our observation, only local smoothing is required to improve the mesh quality and "inflate" the inserted hex sheets, since the trunk is meshed by sweeping, its mesh quality is presumably satisfactory, expensive global smoothing can be avoided and local smoothing is sufficient.

Finally, it is worth mentioning that, after we form a branch, the 3D buffer layer is updated. It erases hexahedra that no longer have at least a face on the boundary and absorbs the hexahedra in the inserted hex sheets that have at least one edge or face on the boundary. This is illustrated in Figure 9(d).

## 8 Creating The Boundary Layer Mesh

After the branches are grafted, the boundary layer mesh can be created by dicing the 3D buffer layer. In the first place, we define a hexahedron on the boundary as following:

**Definition 1.** *A hexahedron is on the boundary if it has at least one vertex, edge or face on the boundary.*

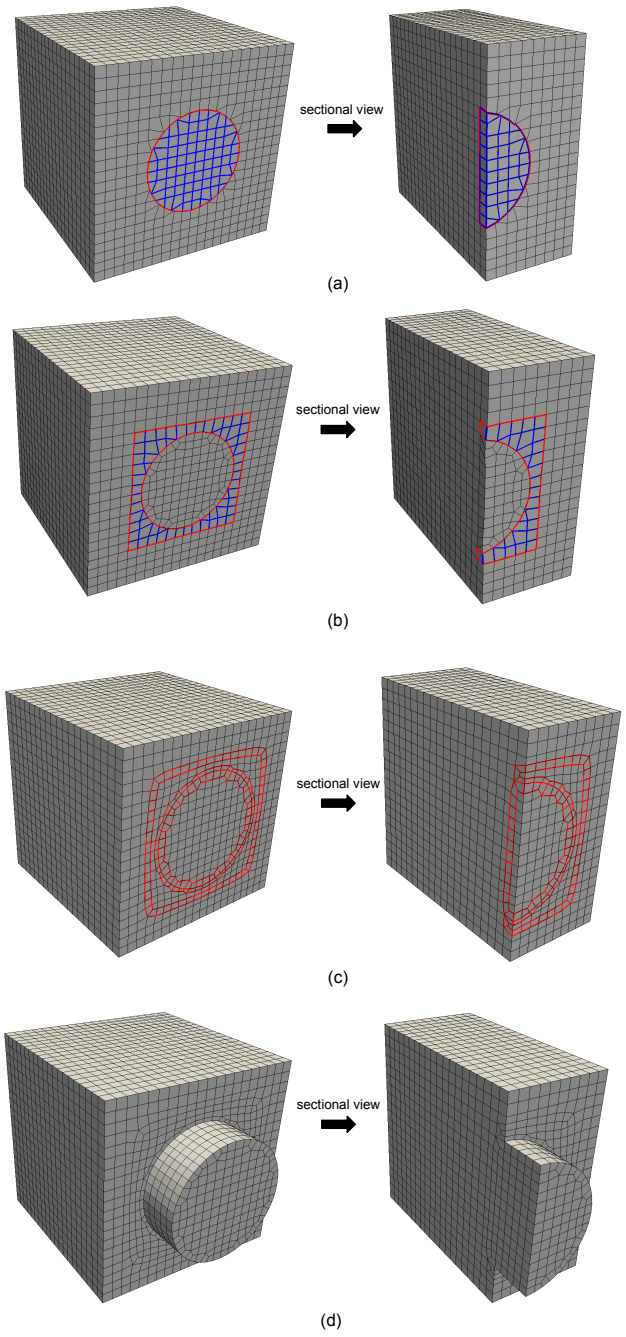


**Fig. 8.** Illustration of improving mesh quality inside and outside a graft loop. The hexahedra highlighted in red represent the inserted hex sheet.

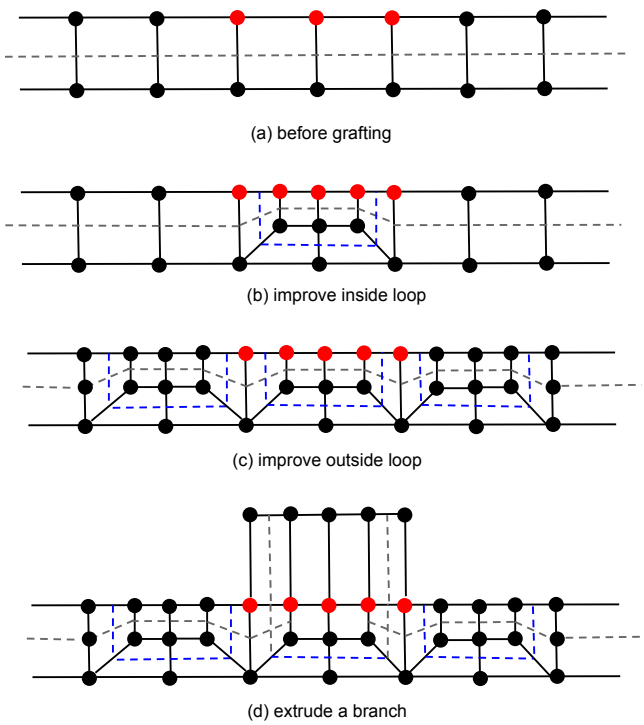


**Fig. 9.** Sectional schematic of the influence of grafting on buffer hex sheets. The red dots represent the points on the graft surface. The grey dash lines represent the buffer hex sheets. The blue dash lines represent the inserted hex sheets to improve mesh quality.

Based on the number of vertices, edges and faces on the boundary, hexahedra on the boundary are classified into several types and they are diced with different templates according to their types. Such classification is shown in Figure 10. After the template for each boundary hex is assigned, the boundary layer mesh can be readily created and the resulting mesh for the example geometry is shown in Figure 11.

However, in the example shown in Figure 11, all the boundaries are simply treated as WALL and other physical boundary conditions (such as FREESTREAM and PERIODIC) are not considered, apart from the target face of a branch extrusion, which is by default considered as FREESTREAM. However the physical boundary conditions have considerable effects on the construction of the boundary layer mesh. If a boundary is PERIODIC or FREESTREAM, then there is no need to create a boundary layer mesh on it.

In order to take FREESTREAM and PERIODIC boundary conditions into account, we need to identify them in the first place. Typically this is done

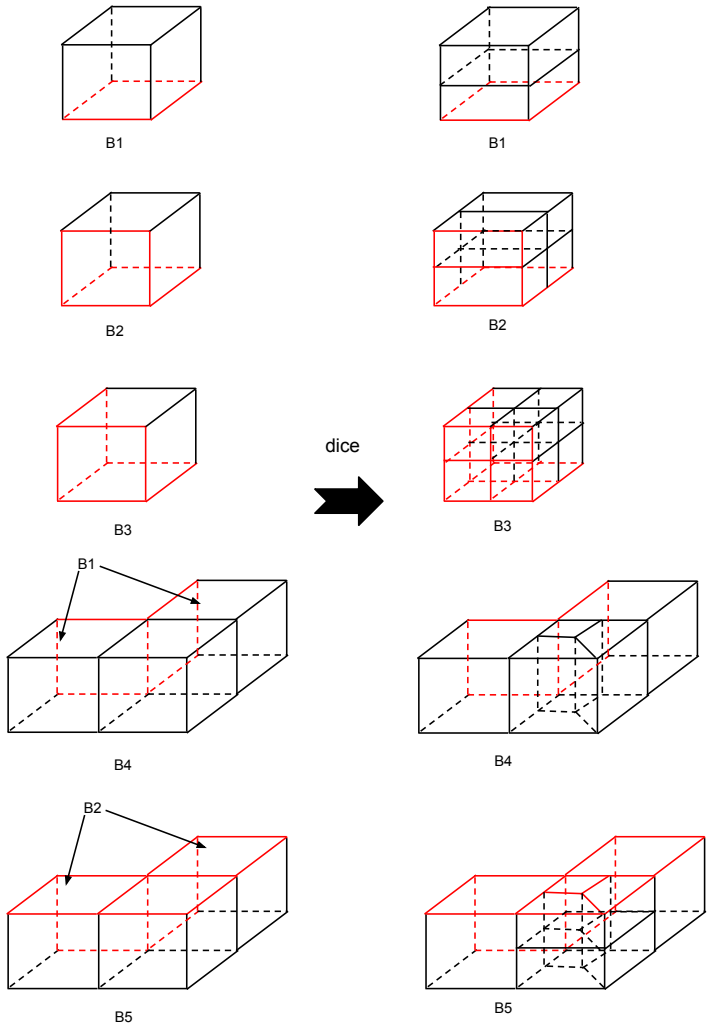


**Fig. 10.** Schematic of dicing templates for boundary hexahedra. The red (dash) lines represent the lines on the boundary.

manually, here we demonstrate this can be achieved automatically with the help of the GA. The PERIODIC boundaries and FREESTREAM boundaries for non-axi-symmetric features can be easily identified. The source face and the target face of the sweep become PERIODIC boundaries and the target faces of extruding the branches become FREESTREAM boundaries. As is shown in Figure 4 (right), axi-symmetric slots are identified when the cavity is extracted, and for the WALL boundaries, they are identified and assigned to

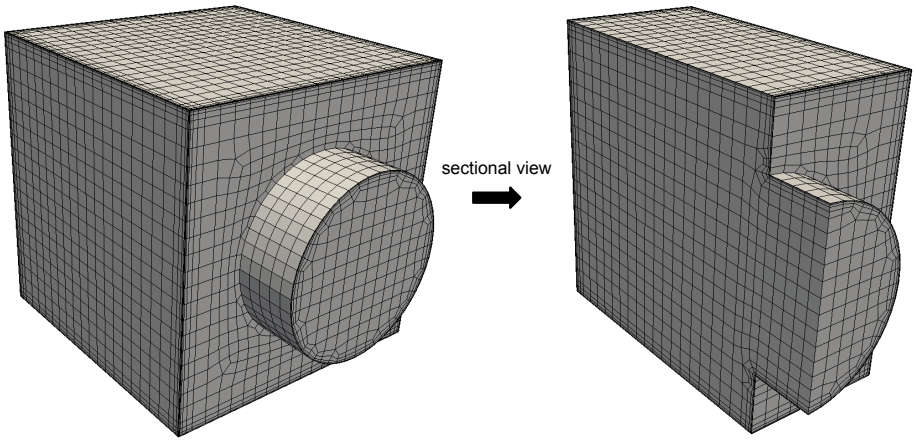


**Fig. 11.** Hex mesh with a boundary layer mesh for the example geometry.

a relevant spool automatically by the GA. When a non-axi-symmetric orifice is grafted on a WALL, a new WALL boundary patch is formed apart from the target face of the extrusion. Based on mechanical consistency, the associated spool for the new WALL patch is forced to be the same as the spool the graft surface is associated to.

After physical boundary conditions are identified, the boundary hex types are re-evaluated based on the number of vertices, edges and faces on the WALL boundary. The boundary layer mesh is then updated by shifting the boundary hex types. As is shown in Figure 12, a B3 can shift to B2, a B2 hexahedron can shift to B1, a B5 can shift to B4 and a B1 can shift to a B4.

## 9 SAS Example

Figure 13 shows the sectional view of the hex mesh(734830 nodes and 713375 hexahedra) with a boundary layer mesh for the extracted cavity(Figure 4). From Figure 13, we can see on the PERIODIC and FREESTREAM boundaries no boundary layer mesh is created, and the thickness of the boundary layer mesh is also adaptive to the local grid spacing on the boundary.

The mesh quality depends heavily on the quality of the quad mesh on the source face, the aspect ratio of the surface grid on base surfaces and the grid spacing along the direction of branch extrusions. The quality of the source face is determined by the Q-morphing algorithm, the aspect ratio of the base surface can be improved by controlling the number of copies in the sweep, and a smooth transition of mesh sizes from the trunk to branches can be easily achieved by automatically sampling the thickness of the buffer layer. Figure 14 shows the quality distribution by the scaled Jacobian metric for the example mesh. The maximum scaled Jacobian is 1.000, the average is 0.981 and the

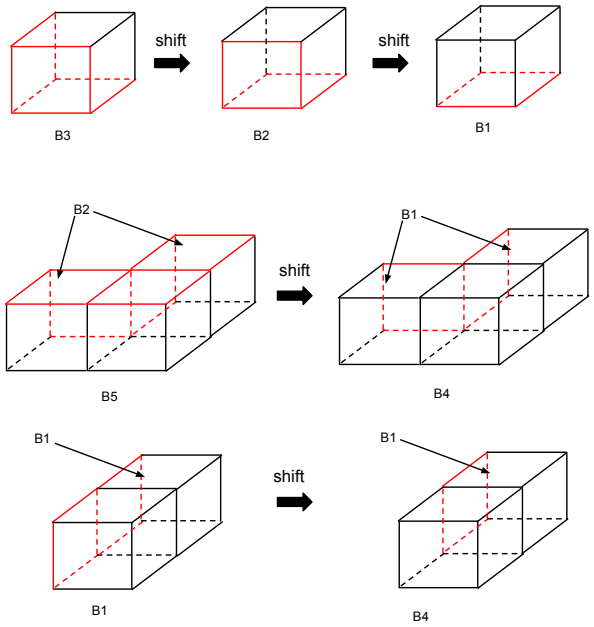


**Fig. 12.** Shift of boundary hex types with consideration of physical boundary conditions. The red (dash) lines represent the lines on the boundaries.

minimum scaled Jacobian is 0.174. Based on our observation, the hexahedral with the lowest quality normally happens around graft loops and are sensitive to the aspect ratio of the surface grid on base surfaces.

Figure 15 illustrates the boundary conditions automatically set up by the meshing process. Since the FREESTREAM boundaries are named after the feature names in the GA and WALL boundaries are associated to the spools by the GA, users only need to provide numeric values for the boundary conditions, such as pressure, temperature and velocity, before completing the mesh setup. If the meshing process is coupled with other low-fidelity analysis processes for the SAS, the whole meshing process could be fully automatic.

## 10 Conclusions and Future Work

In this paper, we have presented a systematic process of producing a quality hex mesh for the turbomachinery SAS. Adaptive 3D boundary layer meshes are automatically created and with the help of the GA physical boundary conditions are also assigned to facilitate the mesh setup in an automated manner. The whole process is automated from the geometry import to the

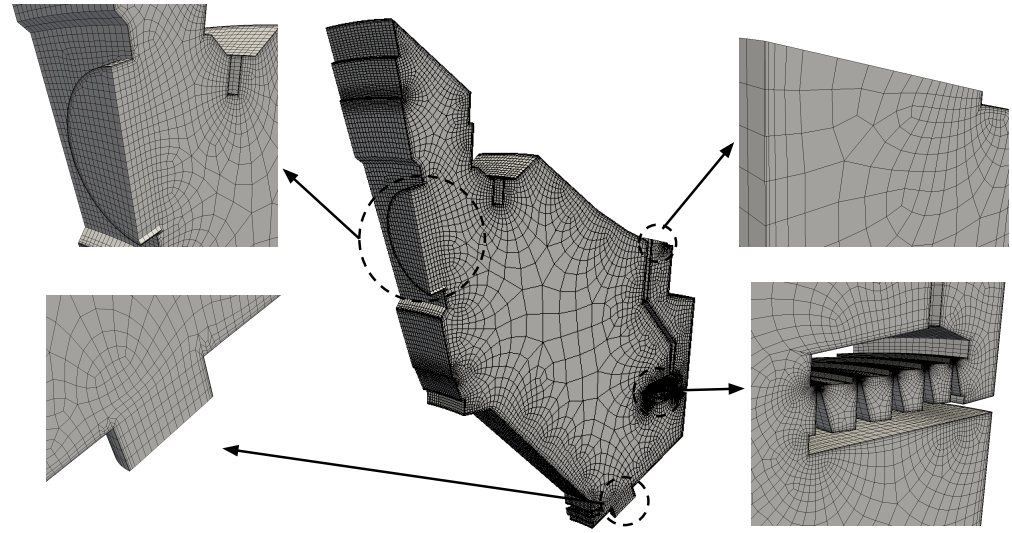

**Fig. 13.** Sectional view of the hex mesh with a boundary layer mesh for the extracted cavity.

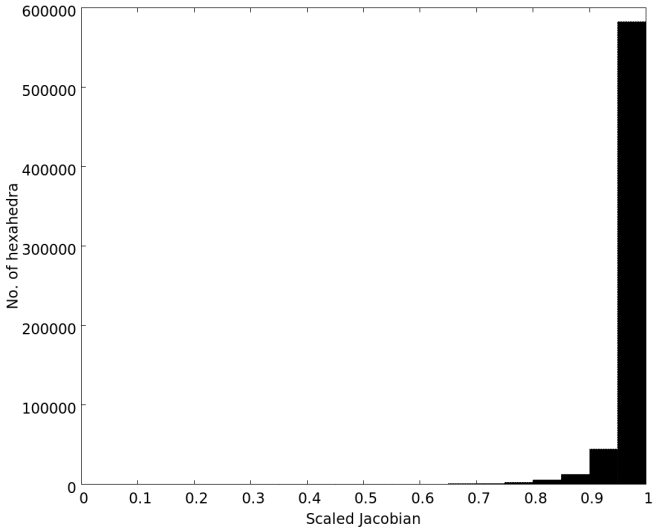


**Fig. 14.** Element quality distribution by the scaled Jacobian metric.

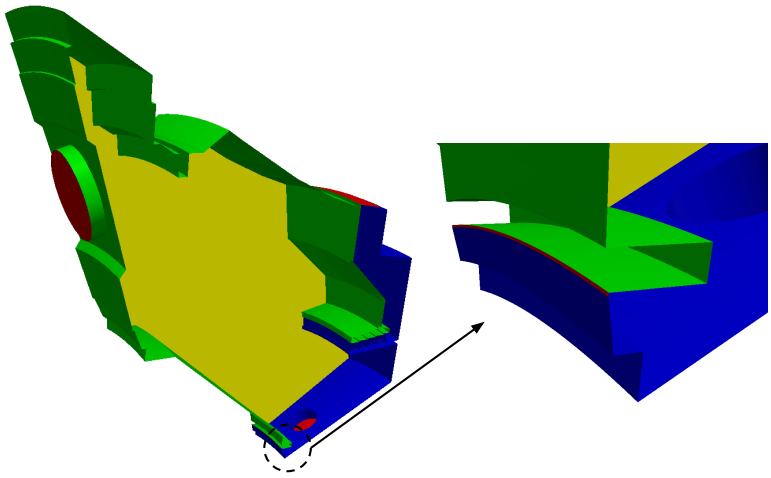

**Fig. 15.** Illustration of boundary conditions automatically set up by the mesh generator. The green patch represents the stationary walls, the blue represents the walls rotating with the IP spool, the red represents free-stream boundaries and the yellow represents the periodic boundaries.

mesh setup and requires minimum human interventions. It has the potential to considerably reduce the time cost in meshing the SAS and hence meets the need of rapid meshing within the design enviroment. Even though the presented hex meshing algorithm is developed for the SAS, it can apply to any 2.5D geometries.

One pitfall of the proposed hex meshing algorithm is that it requires the mesh on the base surface having a sufficient density and reasonable aspect ratio. If the sizes of a branch and the trunk are disparate, the algorithm might produce low quality hexahedra around the graft loop. The future work is to automatically detect such situation and conduct a local refinement on the base surface to improve the local surface mesh density and aspect ratio.

***Acknowledgement***

The authors gratefully acknowledge Rolls-Royce plc for funding this work and granting permission for its publication.

## References

1. Luca Di Mare, Davendu Y.Kulkarni, Feng Wang, et al(2011), Virtual gas turbines: geometry and conceptual description, Proceedings of ASME TurboExpo, Vancouver, Canada.

2. W.N Dawes(2007), Turbomachinery computational fluid dynamics: asymptotes and paradigm shifts, Phil. Trans. R. Soc. A vol.365 no.1859 2553-2585.
3. Jonathan Richard Shewchuk(1998), Tetrahedra mesh generation by Delaunay Refinement, Proceedings of 14th annual symposium on computational geometry.
4. Hang Si, Gartner Klaus(2011), 3D boundary recovery by constrained Delaunay tetrahedralization(2011), Int.J.Nume.Meth.Eng. 85 1341-1364.
5. Feng Wang, Luca Di Mare(2011), On the robust construction of Delaunay Tetrahedralization, Research Notes, 20th International Meshing Roundtable, Paris, France.
6. Jason F Shepherd, Chris R Johnson(2008), Hexahedral mesh generation constraints, Engineering with Computers 24 195-213.
7. Joe E. Thompson, Z.U.A. Warsi and C. Wayne Mastin(1985), Numerical Grid Generation Foundations and Applications, Elsevier Science Publishing Co., Inc.
8. Yongjie Zhang, Thomas J.R. Hughes(2010), Chandrajit L. Bajaj, An automatic 3D mesh generation method for domains with multiple materials, Computer Methods in Applied Mechanics and Engineering, Volume 199, Issues 58, 1, Pages 405-415.
9. WN Dawes(2009), A practical demonstration of scalable, parallel mesh generation, 47th AIAA Aerospace Sciences Meeting Exhibit.
10. http://cubit.sandia.gov/
11. Karl Merkley, Corey Ernst, Jason F. Shepherd, et al, Methods and applications of generalized sheet insertion for hexahedral meshing, Proceedings of 16th International Meshing Roundtable.
12. J.L. Steger, R.L. Sorenson(1979), Automatic mesh-point clustering near a boundary in grid generation with elliptic partial differential equations, Journal of Computational Physics, 33 405-410.
13. Tet D. Blacker, Michael B. Stephenson(1991), Paving: a new approach to automated quadrilateral mesh generation, Int.J.Nume.Meth.Eng. 32 811-847.
14. S. J. Owen, M. L. Staten, S. A. Canann, et al(1999), Q-morphing: an indirect approach to advancing front quad meshing, Int.J.Nume.Meth.Eng. 44 1317-1340.
15. Alper ngr(2009), Off-centers: A new type of Steiner points for computing size-optimal quality-guaranteed Delaunay triangulations, Computational Geometry, 42, 109-118.
16. S. A. Canann, S. N. Muthukrishnan, R. K. Phillips(1998), Topological improvement procedures for quadrilateral finite element meshes, Engineering with Computers 14 168-177.
17. Guy Bunin(2006), Non-local topological clean-up, Proceeding of 15th International Meshing Roundtable.
18. Tian Zhou, Kenji Shimada(2000), An angle-based approach to two-dimensional mesh smoothing, Proceeding of 9th International Meshing Roundtable.
19. Steven R. Jankovich, Steven E. Benzley, Jason F. Shepherd, et al(1999), The graft tool: an all-hexahedral transition algorithm for creating a multi-directional swept volume mesh, 8th International Meshing Roundtable.
20. Scott A. Mitchell, Timothy J. Tautges(1995), Pillowing doublets: refining a mesh to ensure that faces share at most one edge, Proceedings of 4th International Meshing Roundtable.